

```

Spielfeld
- felder: Feld[]
- feldAnzahl: int
+ Spielfeld(feldAnzahl: int, menschen: List<Mensch>, geruechtVerbreiter, geruechtDementierer)
- initialisiereFelder(): void
- platziereMenschen(menschen: List<Mensch>, geruechtVerbreiter, geruechtDementierer): void
+ spieleEineRunde(): void
- menschenBewegen(): void
- menschenUpdate(): void
+ statistik(): Statistik
  
```

```

Feld
- nachbarn: List<Feld>
- menschen: List<Mensch>
+ Feld()
+ addMensch(mensch: Mensch): void
+ addNachbar(nachbar: Feld): void
+ clearMenschen(): void
+ meinungsFindung(): Map<Meinung, Integer>
+ lassMenschenLaufen(): Map<Mensch, Feld>
+ updateMeinungen(): void
  
```

```

<<Enumeration>>
Meinung
HOCHZEIT
KLHOCHZEIT
UNENTSCHL.
KEINE_INFO
G-D-K-I
+ meinungsfindung(meinungen: List<Meinung>): Meinung
  
```

```

Spiel
- anzahlMenschen: int
- anzahlRunden: int
- spielfeld: Spielfeld
+ Spiel(anzahlMenschen: int, anzahlRunden: int, spielfeldGrosse: int)
- erstelleMenschen(anzahlM: int): List<Mensch>
- spielAuswertung(runde: int): void
+ starteSpiel(): void
  
```

```

Mensch
#meinung: Meinung
+ Mensch()
+ waehleNeuesFeld(feldesList<Feld>, feld: Feld): Feld
+ getMeinung(): Meinung
+ setMeinung(meinung: Meinung): void
  
```

```

GeruechtVerbreiter
+ GeruechtVerbreiter(meinung: Meinung): void
+ setMeinung(meinung: Meinung): void

GeruechtDementierer
+ GeruechtDementierer()
+ setMeinung(meinung: Meinung): void
  
```

### Statistik

```

- anzahlHochzeit: int
- anzahlKeineHochzeit: int
- anzahlUnentschlossen: int
+ Statistik(anzahlHochzeit: int, anzahlKeineHochzeit: int, anzahlUnentschlossen: int)
+ getProzentHochzeit(anzahlMenschen: int): double
+ getProzentKeineHochzeit(anzahlMenschen: int): double
+ getProzentUnentschlossen(anzahlMenschen: int): double
  
```